NWCRG
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

I. Swett
Google
M. Montpetit
Triangle Video
V. Roca
INRIA

October 30, 2017

Network Layer Coding for QUIC: Requirements
draft-quic-coding-00

Abstract

This document presents the motivation and requirements for the use of Network Level Packet Erasure Coding to improve the performance of the QUIC protocol that is proposed a new transport protocol. The document does not specify a specific code but lists the salient features that a code should have in order to deal with know loss patterns on QUIC paths.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

In addition, while most of the the terminology in this document is conform to the taxonomy presented in [[NC-Taxonomy]] for clarity and comparison with existing QUIC documents we continue to use the word packet to indication the entity that will be encoded vs. symbol in the taxonomy document.

NOTE: while using drafts in references is not compliant with IETF/ IRTF rules they will be replaced by RFCs as they become available.

2. Introduction

The QUIC (Quick UDP-based Internet Connection)protocol is currently being proposed as new transport protocol than multiplexes connections over UDP. The major elements have been defined and are being implemented by the QUIC IETF working group [QUIC-WG] including wire format, connection establishment, stream multiplexing, stream and connection-level flow control, and data encryption [numerous draft references]. This document addresses an outstanding element of the QUIC protocol, namely how to account and correct for packet losses at

the network layer that will have a very negative impact on transport delay, throughput and reliability. This document presents the salient features and requirements for a network coding (NC) protocol to provide the QUIC packet loss recovery it requires. NC provides a structured, algebraic mechanism to recover lost packets based on a vast heritage of Forward Error Correction (FEC) and has shown better performance of packet recovery than XORs or repetition codes to deal with the losses in the Internet. The Network Coding taxonomy document [[NC-Taxonomy]] contains an overview of top NC concepts. Note: we need a small NC draft that explains how it works.

3. QUIC Background

This section will completed in a future version. For the needs of the current document we need to know that the QUIC packet format contains a unique packet identifier (ID) and a connection ID. Details will be obtained from [[QUIC-Connect]] and [[QUIC-Trans]]

4. Motivation

The QUIC protocol from its early implementations, wanted to address packet losses in the Internet as they can greatly impact protocol performance and impact the performance congestion control mechanisms [[QUIC-Loss]]. For example TCP goodput goes below 20% with 3% loss [are there any other references besides the famous Mathis curve?]. In this section we review the motivations behind the use of a network coding approach to reduce the impact of packet losses on QUIC. It is important to note that we limit the sources of the losses to IP layer and above and losses in lower layers are addressed by other standardization organization.

It is known (is it?) that the main sources of losses in the Internet include (but are not limited to):

- Queuing losses across multiple flows
- Intermittent timeouts
- Connection losses
- Residual physical or MAC layer losses
- Misrouting
- other?

The main feature of all the patterns associated with the loss events above is the fact that losses appear in clusters (burst or correlated losses). Hence they are not the ’random losses’ that can be recovered by non structured mechanisms like XOR or repetitions codes even with high overhead or simple block codes with fixed window sizes. Hence because of the correlated losses, the first requirement for a good code for QUIC is one that allows variable window sizes

that allow to vary with the size of the burst. That will be better suited to recover the losses with statistically approximately the same overhead as the average packet loss without limitations on the loss pattern.

5. Architecture

In order to define the potential NC solution, a detailed architecture is necessary. Hence, the main QUIC/NC architecture topics to be addressed includes the following (each will become a subsection in the future).

- Type of connection: while unicast and/or multicast/broadcast communications are possible over QUIC it is assumed that an initial implementation will be limited to unicast.
- Addressing single source flow or multiple flows: at the the coding level, if there is a multiplexing on top of the coding level, already managed by QUIC machinery, it may be totally transparent. We can assume that individual packets and connections can be individually identified.
- Use of feedback: since the code will need to deal with correlated losses can it benefit from feedback to manage the window as opposed to a fully unidirectional source-destination mechanism. This will allow not to lose any packet part of a current generation before the loss burst ends (we need a reference on window growth and maximum size)
- Minimization of latency:Latency is key for the solution design. This includes reducing extra delay due to encoding/decoding at the ingress, egress and intermediate nodes (middleboxes) especially on delay sensitive paths. At the same time long bandwidth-delay product networks coding should reduce the overall end-to-end delay experienced by an application significantly by minimizing the effect of packet losses and retransmission on TCP congestion control and throughput.
- Throughput aspects: it is expected that the QUIC flows will include high throughput flows, very low throughput flows and mixed sizes flows.
- Interactions with other functionalities: Interactons with congestion control and encryption will also be key. Directions will be taken from [[QUIC-Loss]] and [[QUIC-TLS]] and other relevant documents
- Code changes and future proofing: any protocol designed within QUIC should be able to maybe use more than one code to change codes easily by without major impact this is to address different network conditions or improve performance if a new code was to be developed. It is assumed that the code remain the same for the full QUIC session lifetime but that within a session at least for

the beginning it should be possible to turn off the coding to prevent catastrophic congestion collapse for example.

6. Use-cases

Note: this will be detailed in the next version of the document

7. Requirements

The initial requirements for the QUIC NC are presented below. This list will help to chose the best solution amongst existing codes.

Requirements:

- Simplicity/low complexity: both encoding and decoding operations should be simple and ass little complexity to the QUIC operations; the use of systematic coding will be encouraged.
- Low overhead: the NC overhead to compensate for all losses should be as close as possible to the average loss on the path as to not create additional congestion condition.
- In network coding: there should be ways to create additional coded symbols inside the network either directly or via partial or full decoding.
- Multipath: there should be ways to take advantage of multipath communications for example to send packets and coded symbols on different paths to reduce delay and overhead on some delay or loss sensitive paths.
- Licensing/IPR: the solution should be license/patent free.

8. Next Steps

Besides adding the sections missing in the document based on future discussion it is proposed to define a strawman architecture based on existing codes and using the standard APIs being developed in the RG.

9. IANA Considerations

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

10. Security Considerations

Security: While NC will not impact security in itself it will be important to verify how NC interacts with current encryption used in QUIC and presented in [[QUIC-TLS]].

11. References

11.1. Normative References

[NC-Taxonomy]
Abramson, B. and et. al., "Network Coding Taxonomy", Internet-draft draft-irtf-nwcrg-network-coding-taxonomy-05.txt, July 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <https://www.rfc-editor.org/info/rfc6363>.

11.2. Informative References

[QUIC-Connect]
Roskind, J., "QUIC: Quick UDP Internet Connections", URL https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/preview#.

[QUIC-Loss]
Iyengar, J. and I. Swett, "QUIC Loss Detection and Congestion Control", Internet-draft draft-iyengar-quic-loss-recovery-06.txt, September 2017.

[QUIC-Overview]
"QUIC Overview", URL https://datatracker.ietf.org/wg/quic/about/.

[QUIC-TLS]
Thomson, M. and R. Harrison, "Using Transport Layer Security (TLS) to Secure QUIC", Internet-draft -06.txt, September 2017.

[QUIC-Trans]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Internet-draft draft-ietf-quic-transport-06.txt, September 2017.

Appendix A. Revision information

XXX RFC-Ed please remove this section prior to publication.

Authors' Addresses

Ian Swett
Google
Cambridge, MA
USA

EMail: ianswett@google.com

Marie-Jose Montpetit
Triangle Video
Boston, MA
USA

EMail: marie@mjmontpetit.com

Vincent Roca
INRIA
Grenoble
France

EMail: vincent.roca@inria.fr